STATUS OF CLAIMS:

Claims 1-9 (Cancelled)

10. (Currently Amended). A test kit useful for detecting ~~PS108~~ a polynucleotide in a test sample, said test kit comprising a container containing at least one ~~PS108~~ polynucleotide having at least 50% identity with a sequence selected from the group consisting of SEQUENCE ID NOS: 1-16, and fragments or complements thereof, wherein said fragments have a length of at least 10 nucleotides.

11. (Currently Amended). A purified polynucleotide or fragment thereof ~~derived from a PS108 gene, wherein said polynucleotide is capable of selectively hybridizing to the nucleic acid of said PS108 gene and has~~ having at least 50% identity with a sequence selected from the group consisting of SEQUENCE ID NOS 1-16, and fragments or complements thereof, wherein said fragments have a length of at least 10 nucleotides.

12. (Currently Pending). The purified polynucleotide of claim 11, wherein said polynucleotide is produced by recombinant techniques.

13. (Currently Pending). The purified polynucleotide of claim 11, wherein said polynucleotide is produced by synthetic techniques.

14. (Currently Amended). The purified polynucleotide of claim 11, wherein said polynucleotide comprises a sequence encoding at least one ~~PS108~~ epitope.

15. (Currently Amended). A recombinant expression system comprising a nucleic acid sequence that includes an open reading frame ~~derived from PS108~~ operably linked to a control sequence compatible with a desired host, wherein said nucleic acid sequence has at least 50% identity with a sequence selected from the group consisting of SEQUENCE ID NOS 1-16, and fragments or complements thereof, wherein said fragments have a length of at least 10 nucleotides.

16. (Currently Pending). A cell transfected with the recombinant expression system of claim 15.

Claims 17-22 (Cancelled)

Claims 23-32 (Withdrawn)

33. (Currently Amended). A composition of matter comprising a ~~PS108~~ polynucleotide or fragment thereof, wherein said polynucleotide has at least 50% identity with a polynucleotide selected from the group consisting of

SEQUENCE ID NOS 1-16, and fragments or complements thereof, wherein said fragments have a length of at least 10 nucleotides.

Claim 34 (Withdrawn)

35. (Currently Pending). A test kit of claim 10 further comprising a container with tools useful for collection of said sample, wherein the tools are selected from the group consisting of lancets, absorbent paper, cloth, swabs and cups.

Claims 36-37 (Cancelled)

38. (Twice Amended). A polynucleotide that codes for a ~~PS108~~ protein comprising an amino acid sequence having at least 50% identity to SEQUENCE ID NO 36.

39. (Previously Amended). A polynucleotide comprising DNA having at least 50% identity with SEQUENCE ID NO: 15 or SEQUENCE ID NO:16.

REMARKS

Reconsideration of the above-identified application in view of the foregoing amendments and following arguments is respectfully requested.

Claims 10, 11, 14, 15, 33, 38 and 39 have been amended. No new matter has been added as a result of these amendments. Support for the phrase "wherein said fragments have a length of at least 10 nucleotides" can be found on page 13, line 20 of the specification.

Claim Rejections – 35 U.S.C. Section 112, First Paragraph

Claims 10-16, 33, 35, 38 and 39 are rejected under 35 U.S.C. Section 112, first paragraph, as containing subject matter that was not described in the specification in such a way as to reasonably convey to one skilled in the art that the inventor, at the time the application was filed, had possession of the claimed invention. According to the Examiner, the current claims encompass a genus of nucleic acids that are different than those disclosed in the specification. The Examiner states that the genus includes variants for which no written description is provided in the specification. Applicants respectfully traverse this rejection.

As Applicants discussed in their last Amendment, the inquiry into whether the description requirement is met is determined on a case-by-case basis and is a question of fact. *Manual of Patent Examining Procedure,* Section 2163.04 (8th Edition, February 2003 Revision). When a question regarding the adequacy of the written description arises, the fundamental factual inquiry is whether the specification conveys to those skilled in the art, as of the filing date sought, that Applicant was in possession of the invention being claimed. *Manual of Patent Examining Procedure,* Section 2163.02 (8th Edition, February 2003 Revision). Possession can be shown in a number of ways. For example, an Applicant can show possession by: (1) an actual reduction to practice of the claimed invention; (2) a clear depiction of the invention in detailed drawings or in structural chemical formulas which permit a person skilled in the art to clearly recognize that applicant had possession of the claimed invention; or (3) any description of sufficient, relevant, identifying characteristics so long as a person skilled in the art would recognize that the inventor had possession of the claimed invention. *Manual of Patent Examining Procedure,* Section 2163 (8th Edition, February 2003 Revision).

A description as filed is presumed to be adequate, unless or until sufficient evidence or reasoning to the contrary has been presented by the Examiner to rebut the presumption. *Manual of Patent Examining Procedure,* Section 2163.04 (8th Edition, February 2003 Revision). The Examiner, therefore, must have a reasonable basis to challenge the adequacy of the written description. *Id.* The Examiner has the initial burden of presenting by a preponderance of the evidence why a person skilled in the art would not recognize in an applicants disclosure a description of the invention as defined by the claims. *Id.*

As discussed in their last Amendment, Applicants respectfully submit that the specification as filed is adequate and reasonably conveys to one skilled in the relevant art that the inventors, at the time the application was filed, had possession of the claimed invention. More specifically, with respect to the "50% identity" language, the specification specifically describes how "% identity" (see page 11, lines 30-35 and page 12, lines 1-5) can be determined using various programs known in the art, including the Wisconsin Sequence Analysis Package 8. Applicants herewith enclose the software manual to the Wisconsin Sequence Analysis program, Version 8, which is publicly available from Genetics Computer Group, Madison Wisconsin, as Exhibit A. Support for this submission is found on page 11, line 35 – page 12, line 1. This manual provides the algorithm, parameters, parameter values and other information necessary to, accurately and consistently, calculate the percent identity. This manual indicates on pages 5-21, *inter alia*, that the software used the local homology algorithm of Smith and Waterman (Advances in Applied Mathematics 2:482-489 (1981)). Applicants submit that using the information provided in the specification along with the publicly available software manual that is supplied with the Wisconsin Sequence Analysis program, one skilled in the art would readily be able to discern the nucleic acids encompassed by the scope of the claims and that the inventors were in possession of the claimed invention at the time of filing. Thereupon, Applicants submit that this rejection should be withdrawn.

Rejection of Claims 10-16, 30, 33 and 35 Under 35 U.S.C. Section 102(b)

Claims 10-16, 30, 33 and 35 have been rejected under 35 U.S.C. Section 102(b) as being anticipated by de Louvencourt et al. (U.S. Patent 4,806,472). Applicants respectfully traverse this rejection.

Claims 10, 11, 15 and 33 have been amended to recite that the fragments have a length of at least 10 nucleotides. De Louvencourt et al. does not disclose or suggest a fragment having a length of at least 10 nucleotides. Therefore, because each and every element of the claimed invention is not found in de Louvencourt et al. (U.S. Patent 4,806,472), Applicants submit that this rejection has now been rendered moot and should be withdrawn.

Rejection of Claims 10-14 and 33 Under 35 U.S.C. Section 103(a)

Claims 10-14 and 33 are rejected under 35 U.S.C. Section 103(a) as being unpatentable over Southern (U.S. Patent 6,054,270). Applicants respectfully traverse this rejection.

As discussed in Applicants previous Amendment, in order to establish a *prima facie* case of obviousness, the Examiner must establish three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings (*Manual of Patent Examining Procedure* Section 2142 (8th Edition, February 2003 Revision)). Second, there must be a reasonable expectation of success. *Id.* Finally, the prior art references must teach or suggest all the claim limitations. *Id.* In view of these criteria, Applicants submit that the Examiner has failed to establish a *prima facie* case of obviousness.

The Examiner argues that Southern teaches hybridization of 8-mers to the array to yield double stranded molecules. According to the Examiner, these arrays would inherently and necessarily comprise every 8 mer fragment of SEQ ID NOS 1-16. Claims 10, 11, 15 and 33 have been amended to recite that the fragments have a length of at least 10 nucleotides. Thereupon, Applicants submit that the prior art cited by the Examiner fails to teach or suggest the claims as now amended. Specifically, Southern does not teach any of the sequences of the present invention or fragments thereof.

Therefore, in view of the aforementioned arguments, Applicants submit that the rejection of claims 10-14 and 33 under 35 U.S.C. Section 103(a) as being unpatentable over Southern should be withdrawn.

Applicants submit that the claims are now in condition for allowance.

Should the Examiner have any questions concerning the above, she is respectfully requested to contact the undersigned at the telephone number listed below. If any additional fees are incurred as a result of the filing of this paper, authorization is given to charge Deposit Account No. 23-0785.

Respectfully submitted,

By: ____________________
Lisa V. Mueller (Reg. No. 38,978)
Attorney for Applicant

WOOD, PHILLIPS, KATZ, CLARK & MORTIMER
500 MADISON STREET, SUITE 3800
CHICAGO, IL 60661
(312) 876-1800

CERTIFICATE OF MAILING

I hereby certify that this amendment is being hand delivered to the Commissioner of Patents, PO Box 1450, Alexandria, VA 22313-1450 on September 29, 2003.

Kim Annel

EXHIBIT A

FUNCTION

BestFit makes an optimal alignment of the best segment of similarity between two sequences. Optimal alignments are found by inserting gaps to maximize the number of matches using the *local homology* algorithm of Smith and Waterman.

DESCRIPTION

BestFit inserts gaps to obtain the optimal alignment of the best region of similarity between two sequences, and then displays the alignment in a format similar to the output from Gap. The sequences can be of very different lengths and have only a small segment of similarity between them. You could take a short RNA sequence, for example, and run it against a whole mitochondrial genome.

SEARCHING FOR SIMILARITY

BestFit is the most powerful method in the Wisconsin Sequence Analysis Package™ for identifying the best region of similarity between two sequences whose relationship is unknown.

EXAMPLE

The sequence gamma.seq contains an Alu family sequence somewhere in the first 500 bases. alu.seq contains a generic human Alu family repeat. The two sequences are aligned and the best segment of similarity is found with BestFit.

```
% bestfit

 BESTFIT of what sequence 1 ?  gamma.seq

                   Begin (* 1 *) ?
                 End (* 11375 *) ?  500
                Reverse (* No *) ?

 to what sequence 2 (* gamma.seq *) ?  alu.seq

                   Begin (* 1 *) ?
                 End (*   207 *) ?
                Reverse (* No *) ?

 What is the gap creation penalty (* 5.00 *) ?

 What is the gap extension penalty (* 0.30 *) ?

 What should I call the paired output display file (* gamma.pair *)

 Aligning ............

              Gaps:      3
           Quality:  129.3
     Quality Ratio:  0.625
      % Similarity: 84.466
            Length:    209

%
```

OUTPUT

Here is the output file. Notice how BestFit finds and displays only the best segments of similarity:

```
 BESTFIT of: gamma.seq  check: 6474  from: 1  to: 500

Human fetal beta globins G and A gamma
from Shen, Slightom and Smithies,  Cell 26; 191-203.
Analyzed by Smithies et al. Cell 26; 345-353.

 to: alu.seq  check: 4238  from: 1  to: 207

HSREP2 from the EMBL data library
Human Alu repetitive sequence located near the insulin gene
Dhruva D.R., Shenk T., Subramanian K.N.; "Integration in vivo into
Simian virus 40 DNA of a sequence that resembles a certain family of
genomic interspersed repeated sequences"; Proc. Natl. Acad. Sci. USA
77:4514-4518(1980). . . .

 Symbol comparison table: Gencoredisk:[Gcgcore.Data.Rundata]Swgapdna.Cmp
 CompCheck: 5234

         Gap Weight:  5.000      Average Match:  1.000
      Length Weight:  0.300   Average Mismatch: -0.900

            Quality:  129.3             Length:    209
              Ratio:  0.625               Gaps:      3
 Percent Similarity: 84.466   Percent Identity: 84.466

 gamma.seq x alu.seq       June 20, 1994 15:15  ..

                  .         .         .         .         .
     137 AGACCAACCTGGCCAACATGGTGAAATCCCATCTCTAC.AAAAATACAAA 185
         |||||| |||||||||||||||||||  ||||||||||  ||||||||||
       1 AGACCAGCCTGGCCAACATGGTGAAACTCCATCTCTACTGAAAATACAAA 50

                  .         .         .         .         .
     186 AATTAGACAGGCATGATGGCAAGTGCCTGTAATCCCAGCTACTTGGGAGG 235
         |||||| |||||||| ||    ||||||| |||||||||||||| |||||
      51 AATTAGCCAGGCATGGTGATGCGTGCCTGGAATCCCAGCTACTTAGGAGG 100

                  .         .         .         .         .
     236 CTGAGGAAGGAGAATTGCTTGAACCTGGAAGGCAGGAGTTGCAGTGAGCC 285
         |||||  || |||||  ||| ||||  | |||  |  |||||||||||||
     101 CTGAGACAGAAGAATCCCTTAAACCAAG.AGGTGGAGGTTGCAGTGAGCC 149

                  .         .         .         .         .
     286 GAGATCATACCACTGCACTCCAGCCTGGGTGACAGAACAAGACTCTGTCT 335
         ||||||  ||  |||||||||||||| ||||||||| | ||||||  |||
     150 GAGATCGCACGGCTGCACTCCAGCCT.GGTGACAGAGCGAGACTCCATCT 198

     336 CAAAAAAAA 344
         |||||||||
     199 CAAAAAAAA 207
```

RELATED PROGRAMS

When you want an alignment that covers the whole length of both sequences, use Gap. When you are trying to find only the best segment of similarity between two sequences, use BestFit. PileUp creates a multiple sequence alignment of a group of related sequences, aligning the whole length of all sequences. DotPlot displays the entire surface of comparison for a comparison of two sequences. GapShow displays the pattern of differences between two aligned sequences. PlotSimilarity plots the average similarity of two or more aligned sequences at each position in the alignment. Pretty displays alignments of several sequences. LineUp is an editor for editing multiple sequence alignments. CompTable helps generate scoring matrices for peptide comparison.

ALGORITHM

BestFit uses the *local homology* algorithm of Smith and Waterman (Advances in Applied Mathematics 2; 482-489 (1981)) to find the best segment of similarity between two sequences. BestFit reads a scoring matrix that contains values for every possible GCG symbol match (see the LOCAL DATA FILES topic below). The program uses these values to construct a path matrix that represents the entire surface of comparison with a score at every position for the best possible alignment to that point. The *quality* score for the best alignment to any point is equal to the sum of the scoring matrix values of the matches in that alignment, less the gap creation penalty times the number of gaps in that alignment, less the gap extension penalty times the total length of all gaps in that alignment. The gap creation and gap extension penalties are set by you. If the best path to any point has a negative value, a zero is put in that position.

After the path matrix is complete, the highest value on the surface of comparison represents the end of the best region of similarity between the sequences. The best path from this highest value backwards to the point where the values revert to zero is the alignment shown by BestFit. This alignment is the best segment of similarity between the two sequences.

For nucleic acids, the default scoring matrix has a *match* value of 1.0 for each identical symbol comparison and -0.90 for each non-identical comparison (not considering nucleotide ambiguity symbols for this example). The *quality* score for a nucleic acid alignment can, therefore, be determined using the following equation:

$$\begin{aligned}\mathrm{Quality} = {} & 1.0 \times \mathrm{TotalMatches} + -0.90 \times \mathrm{TotalMismatches} \\ & - (\mathrm{GapCreationPenalty} \times \mathrm{GapNumber}) \\ & - (\mathrm{GapExtensionPenalty} \times \mathrm{TotalLengthOfGaps})\end{aligned}$$

The *quality* score for a protein alignment is calculated in a similar manner. However, while the default nucleic acid scoring matrix has a single value for all non-identical comparisons, the default protein scoring matrix has different values for the various non-identical amino acid comparisons. The *quality* score for a protein alignment can therefore be determined using the following equation (where Total_{AA} is the total number of A-A (Ala-Ala) matches in the alignment, CmpVal_{AA} is the value for an A-A comparison in the scoring matrix, Total_{AB} is the total number of A-B (Ala-Asx) matches in the alignment, CmpVal_{AB} is the value for an A-B comparison in the scoring matrix, ...) :

$$\begin{aligned}\mathrm{Quality} = {} & \mathrm{CmpVal}_{AA} \times \mathrm{Total}_{AA} \\ & + \mathrm{CmpVal}_{AB} \times \mathrm{Total}_{AB} \\ & - \mathrm{CmpVal}_{AC} \times \mathrm{Total}_{AC} \\ & \quad \vdots \\ & - \mathrm{CmpVal}_{ZZ} \times \mathrm{Total}_{ZZ} \\ & - (\mathrm{GapCreationPenalty} \times \mathrm{GapNumber}) \\ & - (\mathrm{GapExtensionPenalty} \times \mathrm{TotalLengthOfGaps})\end{aligned}$$

For a more complete discussion of scoring matrices, see the **Data Files** manual.

CONSIDERATIONS

BestFit Always Finds Something

BestFit always finds an alignment for any two sequences you compare – even if there is no significant similarity between them! You must evaluate the results critically to decide if the segment shown is not just a random region of relative similarity.

The Segments Shown Obscure Alternative Segments

BestFit only shows one segment of similarity, so if there are several, all but one is obscured. You can approach this problem with graphic matrix analysis (see the Compare and DotPlot programs). Alternatively, you can run BestFit on ranges outside the ranges of similarity found in earlier runs to bring other segments out of the shadow of the best segment.

The Best Fit is Only One Member of a Family

Like all fast gapping algorithms, the alignment displayed is a member of the family of best alignments. This family may have other members of equal quality, but will not have any member with a higher quality. The family is usually significantly different for different choices of gap creation and gap extension penalties. See the CONSIDERATIONS topic in the entry for the Gap program in the **Program Manual** to learn more about how to assign gap creation and gap extension penalties.

The Surface of Comparison

The magnitude of the computer's job is proportional to the area of the surface of comparison. That area is determined by the product of the lengths of the two sequences compared. BestFit can evaluate a surface of up to 3.5 million elements. This surface would be large enough to compare two sequences approximately 1,870-symbols long, or one sequence 200-symbols long with another sequence 17,500-symbols long. When you have much longer sequences that are known to align well, you can use the command-line option `-LIMit` to use the surface more efficiently.

The Public Scoring Matrix for Nucleic Acid Comparisons is Very Stringent

The scoring matrix swgapdna.cmp penalizes mismatches -0.9 so the segments found may be very brief. This penalty means that the alignment cannot be extended by three bases to pick one extra match. The scoring matrix used by Smith and Waterman, when local alignments were first described, used -0.333 for the mismatch penalty. You can use Fetch to copy randomdna.cmp and rename it swgapdna.cmp to use these values, or use nwsgapdna.cmp, which has no mismatch penalty at all.

Rapid Alignment

When possible, BestFit tries to find the optimal alignment very quickly. If this rapid alignment is not unambiguously optimal, BestFit automatically realigns the sequences to calculate the optimal alignment. When this occurs, the monitor of alignment progress on your terminal screen (`Aligning . . .`) is displayed twice for a single alignment.

ALIGNING LONG SEQUENCES

This program can align very long sequences if you know roughly where the alignment of interest begins. Run the program with the command line option `-LIMit`. Then set the starting coordinates for each sequence near the point where the alignment of interest begins and set gap shift limits on each sequence. The program then aligns the sequences from your starting point such that the sequences do not get out of phase by more than the gap shift limits you have set. If you started both sequences at base number one and set the gap shift limit for sequence one to 100 and for sequence two to 50, then base 350 in sequence one could not be gapped to any base outside of the range from 300 to 450 on sequence two.

If you omit `-LIMit` on the command line, the program automatically sets gap shift limits if they are needed to allow the alignment of long sequences to proceed. In this case, the program limits the total length of gaps that can be inserted into each sequence and calculates the best alignment within this incomplete, or *limited*, surface of comparison. The program then performs a calculation to determine whether the alignment could possibly be improved if there were no restriction on the total length of gaps in each sequence. If the program cannot rule out this possibility, it displays the message `*** Alignment is not guaranteed to be optimal ***`. Because the criteria used in the calculation for guaranteeing an optimal alignment are very stringent, a limited alignment often may be optimal even if this message is displayed. In any event, the program continues to completion.

EVALUATING ALIGNMENT SIGNIFICANCE

This program can help you evaluate the significance of the alignment, using a simple statistical method, with the `-RANdomizations` command line option. The second sequence is repeatedly shuffled, maintaining its length and composition, and then realigned to the first sequence. The average alignment score, plus or minus the standard deviation, of all randomized alignments is reported in the output file. You can compare this average *quality* score to the quality score of the actual alignment to help evaluate the significance of the alignment. The number of randomizations can be specified along with the `-RANdomizations` command line qualifier; the default is 10.

The score of each randomized alignment is reported to the screen. You can use `<Ctrl>C` to interrupt the randomizations and output the results from those randomized alignments that have been completed.

By ignoring the statistical properties of biological sequences, this simple Monte Carlo statistical method may give misleading results. Please see Lipman, D.J, Wilbur, W.J., Smith, T.F., and Waterman, M.S. (Nucl. Acids Res. 12; 215-226 (1984)) for a discussion of the statistical significance of nucleic acid similarities.

ALIGNMENT METRICS

BestFit and Gap display four figures of merit for alignments: Quality, Ratio, Identity, and Similarity.

The Quality (described above) is the metric maximized in order to align the sequences. Ratio is the quality divided by the number of bases in the shorter segment. Percent Identity is the percent of the symbols that actually match. Percent Similarity is the percent of the symbols that are similar. Symbols that are across from gaps are ignored. A similarity is scored when the scoring matrix value for a pair of symbols is greater than or equal to 0.50, the *similarity threshold*. This threshold is also used by the display procedure to decide when to put a ':' (colon) between two aligned symbols. You can reset it from the command line with the second optional parameter of `-PAIr`. For instance, the expression `-PAIr=1.0,0.5` would set the similarity threshold to 0.5.

The similarity and identity metrics are not optimized by alignment programs so they should not be used to compare alignments.

PEPTIDE SEQUENCES

If your input sequences are peptide sequences, this program uses a scoring matrix with matches scored as 1.5 and mismatches scored according to the evolutionary distance between the amino acids as measured by Dayhoff and normalized by Gribskov (Gribskov and Burgess Nucl. Acids Res. 14(16); 6745-6763 (1986)).

RESTRICTIONS

Input sequences may not be more than 30,000-symbols long. This program cannot evaluate a surface of comparison larger than 5.5 million elements. A 200 x 27,500 comparison is possible, as well as a 2,300 x 2,300 comparison. See the ALIGNING LONG SEQUENCES topic for help in aligning long sequences that would normally exceed the maximum surface of comparison. You can also ask your system manager to increase the maximum surface of comparison if your system has enough virtual memory.

SEQUENCE TYPE

The function of BestFit depends on whether your input sequence(s) are protein or nucleotide. Normally the type of a sequence is determined by the presence of either `Type: N` or `Type: P` on the last line of the text heading just above the sequence itself. If your sequence(s) are not the correct type, turn to Appendix VI for information on how to change or set the type of a sequence.

COMMAND-LINE SUMMARY

All parameters for this program may be put on the command line. Use the option `-CHEck` to see the summary below and to have a chance to add things to the command line before the program executes. In the summary below, the capitalized letters in the qualifier names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose qualifiers or parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Basic Concepts: Using Programs in the **User's Guide**.

```
Minimal Syntax: % bestfit [-INfile1=]gamma.seq  [-INfile2=]alu.seq -Default

Prompted Parameters:

-BEGin1=1 -BEGin2=1      beginning of each sequence
-END1=500 -END2=207      end of each sequence
-NOREV1     -NOREV2      strand of each sequence
-GAPweight=5.0           gap creation penalty    (3.0 is protein default)
-LENgthweight=0.3        gap extension penalty   (0.1 is protein default)
[-OUTfile1=]gamma.pair   output file for alignment

Local Data Files: -DATa=swgapdna.cmp  scoring matrix for nucleic acids
                  -DATa=swgappep.cmp  scoring matrix for peptides

Optional Parameters:

-OUTfile2=gamma.gap     new sequence file for sequence 1 with gaps added
-OUTfile3=alu.gap        "    "         "  "     "     2  "  "    "
-LIMit1=499 -LIMit2=206 limit the surface of comparison
-RANdomizations[=10]    determine average score from 10 randomized
                            alignments
-PAIr=1.0,0.5,0.1       thresholds for displaying '|', ':', and '.'
-WIDth=50               the number of sequence symbols per line
-PAGe=60                adds a line with a form feed every 60 lines
-NOBIGGaps              suppresses abbreviation of large gaps with '.'s
-HIGhroad               makes the top alignment for your parameters
-LOWroad                makes the bottom alignment for your parameters
-NOSUMmary              suppresses the screen summary
```

ACKNOWLEDGEMENTS

Gap and BestFit were originally written for Version 1.0 by Paul Haeberli from a careful reading of the Needleman and Wunsch (J. Mol. Biol. 48; 443-453 (1970)) and the Smith and Waterman (Adv. Appl. Math. 2; 482-489 (1981)) papers.

Limited alignments were designed by Paul Haeberli and added to the Package for Version 3.0. They were united into a single program by Philip Delaquess for Version 4.0. Default gap penalties for protein alignments were modified according to the suggestions of Rechid, Vingron and Argos (CABIOS 5; 107-113 (1989)).

LOCAL DATA FILES

The files described below supply auxiliary data to this program. The program automatically reads them from a public data directory unless you either 1) have a data file with exactly the same name in your current working directory; or 2) name a file on the command line with an expression like `-DATa1=myfile.dat`. For more information see Chapter 4, Using Data Files in the **User's Guide.**

If the first sequence you name is a nucleic acid, BestFit uses the scoring matrix in the public file swgapdna.cmp. (SW stands for Smith and Waterman.) If the first sequence you name is a peptide sequence, BestFit reads swgappep.cmp instead. The presence of these files in your current working directory causes BestFit to read your version instead. (See the **Data Files** manual for more information about scoring matrices.)

OPTIONAL PARAMETERS

The parameters and switches listed below can be set from the command line. For more information, see "Using Program Parameters" in Chapter 3, Basic Concepts: Using Programs in the **User's Guide.**

`-LIMit1=20 and -LIMit2=20`

let you set *gap shift limits* for each sequence. When you already know of a long similarity between two sequences you can "zip" them together using this mode. The beginning coordinates for each sequence must be near the beginning of the alignment you want to see. The alignment continues so that gaps inserted do not require the sequences to get out of step by more than the gap shift limits. You can align very long sequences rapidly. The surface of comparison is still limited to 3.5 million. The size of a comparison can be predicted by multiplying the average length of the two sequences by the sum of the two shift limits.

If you add `-LIMit` to the command line without any qualifier value, the program prompts you to enter gap shift limits for each sequence.

`-RANdomizations=10`

reports the average alignment score and standard deviation from 10 randomized alignments in which the second sequence is repeatedly shuffled, maintaining the length and composition of the original sequence, and then aligned to the first sequence. You can use the optional parameter to set the number of randomized alignment to some number other than 10.

`-OUTfile2=seqname1.gap -OUTfile3=seqname2.gap`

This program can write three different output files. The first displays the alignment of sequence one with sequence two. The second is a new sequence file for sequence one, possibly expanded by gaps to make it align with sequence two. The third, like the second, is a new sequence file for sequence two, possibly expanded by gaps to make it align with sequence one. The program writes only the first file unless there are output file options on the command line. If there are any output files named on the command line, *only* those output files are written. If you add

-OUT to the command line without any qualifying filename, then the program will write the second and third output files after prompting you for their names.

Aligned sequences (in sequence files) can be displayed with GapShow. Their similarity can be displayed with PlotSimilarity.

`-PAIr=1.0,0.5,0.1`

The paired output file from this program displays sequence similarity by printing one of three characters between similar sequence symbols: a pipe character(|), a colon (:), or a period (.). Normally a pipe character is put between symbols that are the same, a colon is put between symbols whose comparison value is greater than or equal to 0.50, and a period is put between symbols whose comparison value is greater than or equal to 0.10. You can change these *match display thresholds* from the command line. The three parameters for -PAIr are the display thresholds for the pipe character, colon, and period. The match display criterion for a pipe character changes from symbolic identity (the default) to the quantitative threshold you have set in the first parameter. A pipe character will no longer be inserted between identical symbols unless their comparison values are greater than or equal to this threshold. If you still want a pipe character to connect identical symbols, use x instead of a number as the first parameter. (See the **Data Files** manual for more information about scoring matrices.)

`-PAGe=64`

When you print the output from this program, it may cross from one page to another in a frustrating way -- especially when you print on individual sheets. This option adds form feeds to the output file in order to try to keep clusters of related information together. You can set the number of lines per page by supplying a number after the -PAGe qualifier.

`-WIDth=50`

puts 50 sequence symbols on each line of the output file. You can set the width to anything from 10 to 150 symbols.

`-NOBIGGaps`

suppresses large gap abbreviations, showing all the sequence characters across from large gaps. Usually, gaps that extend one sequence by more than one complete line of output are abbreviated with three dots arranged in a vertical line.

`-LOWroad` and `-HIGhroad`

The insertion of gaps is, in many cases, arbitrary, and equally optimal alignments can be generated by inserting gaps differently. When equally optimal alignments are possible, this program can insert the gaps differently if you select either the -LOWroad or the -HIGhroad options. Here are examples for the alignment of GACCAT with GACAT with different parameters.

```
For:         Match = 1.0        MisMatch = -0.9
       Gap weight = 1.0   Length Weight =  0.0

LowRoad:     1 GACCAT 6
               .  |||      Quality = 4.0
             1 GA.CAT 5

HighRoad:    1 GACCAT 6
               ||| ||      Quality = 4.0
             1 GAC.AT 5
```

```
For:         Match = 1.0        MisMatch =  0.0
        Gap weight = 3.0   Length Weight =  0.0

HighRoad:    1 GACCAT 6
                |||            Quality = 3.0
             1 GACAT. 5

LowRoad:     1 GACCAT 6
                  |||          Quality = 3.0
             1 .GACAT 5
```

Essentially the *low road* shifts all of the arbitrary gaps in sequence two to the left and all of the arbitrary gaps in sequence one to the right. The *high road* does exactly the opposite. When neither *high road* nor *low road* is selected, the program tries not to insert a gap whenever that is possible and uses the high road alternative for all collisions.

`-SUMmary`

writes a summary of the program's work to the screen when you've used the `-Default` qualifier to suppress all program interaction. A summary typically displays at the end of a program run interactively. You can suppress the summary for a program run interactively with `-NOSUMmary`.

Use this qualifier also to include a summary of the program's work in the log file for a program run in batch.

Printed: July 13, 1995 08:19 (1162)